REMARKS

Applicant acknowledges with thanks the examiner's indication that claims 5, 7, 10-19, 24, 26 and 29-38 would be allowable.

Applicant amended independent claim 1 to clarify that the processor is a multi-threaded processor and that the processing thread is one of multiple threads processing on the multi-threaded processor. Applicant similarly amended independent claim 20. Additionally, applicant amended claims 6 and 25 to make the language recited therein consistent with the amended language of the respective independent claims.

The examiner rejected claims 1-4, 6, 8-9, 20-23, 25 and 27-28 under 35 USC § 103(a) as being unpatentable by U.S. Patent No. 6,279,066 to Velingker.

Specifically, the examiner stated:

> 9. Referring to claim 1, Velingker has taught a method of operating a processor comprising:
> a) receiving data specified by execution of a fast-write instruction in a processing stream identified by a processing stream number. See Figs.2-4 and column 5, lines 19-50. Note from Fig.3 that multiple processors each execute different streams of instructions (threads 1, 2, and 3). When a stream wants access to a shared resource, it issues an instruction (fast-write instruction) to set a bit in the RNC (register).
> b) the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with one of multiple streams available on the processor. See Figs.2-3, column 5, lines 19-50, and column 6, lines 10-18. In one embodiment of the invention, the resource negotiation cell (RNC) is a register with three groups of bits, each group comprising two bits (a request bit and a completion bit). Each group is associated with a different stream. When a stream wants to request a resource or indicate access completion, a fast-write instruction including data is issued to set an appropriate bit in the corresponding group.
> c) selecting a group of bits associated with the processing stream, the group of bits being selected from the multiple groups of bits of the register specified by the fast-write instruction according to the processing stream number. Again, see column 5, lines 19-50, column 6, lines 10-18, and Figs.2-3. When a stream wants to access a shared resource or indicate access completion, the stream's group of bits is selected for modification.
> d) loading the data into the selected bit positions of the register. See column 5, lines 19-50, column 6, lines 10-18, and Figs.2-3. When the stream wants to access a shared resource, it will write a '1' into a bit of

> the group of bits. And, when it is finished accessing the resource, it will write a '1' into another bit of the group of bits.
> e) while Velingker has not explicitly taught that each stream executed by the processors in the multiprocessor system constitutes a thread. Official Notice is taken that in a multiprocessor system, each processor may be configured to execute a different thread. Further known are the advantages of multithreading. One such advantage is that one when thread is stalled, another thread may continue executing, as they are independent streams With only a single thread, if the thread stalls, all resources stall. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Velingker's streams to be threads such that each processor executes a different thread. (Office Action, pages 3-5, paragraph 9)

Applicant's independent claim 1 recites "receiving data specified by execution of a fast-write instruction in one of multiple threads processing on the multi-threaded processor, the one of the multiple threads identified by a processing thread number, the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with a corresponding thread of the multiple threads processing on the multi-threaded processor."

Velingker describes a resource negotiation mechanism to arbitrate between a plurality of accessing processing agents for access to a common resource such as a register or a memory location (Abstract, col. 1, lines 15-18). Velingker's apparatus includes a shared resource negotiator (SRN) 100 and three processors (102, 104 and 106) seeking to access a common resource such as shared control register (FIG. 1, and col. 3, lines 49-52). Velingker explains that:

> **In particular, FIG. 1 shows a shared resource negotiator 100 including three access request bits 160, 162, 164 for writing by each of the three processors 102-106, and a corresponding three grant bits 170, 172, 174 for reading by the same three processors 102-106. The processors 102, 104 and 106 include appropriate software which allows the respective processor to access the shared control register 150 only after first requesting access to the shared control register 150 by writing to the corresponding request bit 160, 162 or 164 and reading a successful grant value in the corresponding grant bit 170, 172 or 174. (Col. 3, lines 55-65)**

Velingker further explains:

> **FIG. 3 shows that according to the present embodiment, each of the processors 102-106 can write only to its assigned bit location within the shared resource negotiator 100 and not to the other bit locations in the shared resource negotiator 100. For example, the first processor 102 can write only to bit 0 and not to bits 1 and 2 of the shared resource negotiator**

> **100, the second processor 104 can write only to bit 1 and not to bits 0 and 2, and the third processor 106 can write only to bit 2 and not to bits 0 and 1. However, in the disclosed embodiment, all three of the processors 102-106 can read all bits of the shared resource negotiator 100, providing an additional source of information to the accessing processors 102-106 about the granting of access to the corresponding shared resource to other processors as well as to the corresponding processor.**
>
> **FIG. 4 shows a process of requesting permission to write to a shared resource, e.g., the shared control register 150 shown in FIG. 1, in accordance with the principles of the present invention.**
>
> **In particular, FIG. 4 shows in step 402 that a processor desiring to write to a shared resource first lodges a request for the shared resource by writing a predetermined logic level, e.g., a `1', to its assigned request bit in the corresponding shared resource negotiator 100.**
>
> **In step 404, the requesting processor reads back the grant status (i.e., access granted or access denied) from the assigned read bit in the shared resource negotiator 100. In the disclosed embodiment the write and read bit are in the same relative location in the shared resource negotiator 100 register.**
>
> **Step 406 determines whether or not the grant bit read back in step 404 indicates access granted (e.g., `1') or access denied (e.g., `0'). If the requesting processor reads back a logic value indicating that access has been granted, e.g., a `1', the requesting processor will then presume that it has been granted permission to write to the shared resource and complete its access to the shared resource in step 408. In the disclosed embodiment, the processor granted permission to access the shared resource will retain that permission until the corresponding request bit is cleared, e.g., by a write from the corresponding processor. (Col. 5, lines 26-67)**

Thus, a processor from one of Velingker's processors 102-106 that is to access a common resource makes a request for that resource by setting a corresponding request bit on the SRN 100. Velingker, however, does not describe anywhere that any of its processors is a multi-threaded processor on which multiple threads are processing. Indeed, Velingker does not even mention threads. Velingker certainly does not describe a fast-write instruction in one of multiple threads processing on a multi-threaded processor that specifies a register having multiple groups of bits, each of which being associated with a different one of the multiple threads on the multi-threaded processor. Accordingly, Velingker fails to disclose or suggest at least the feature of "receiving data specified by execution of a fast-write instruction in one of multiple threads processing on the multi-threaded processor, the one of the multiple threads identified by a processing thread number, the fast-write instruction further specifying a register, the register

having multiple groups of bits, each group of bits associated with a corresponding thread of the multiple threads processing on the multi-threaded processor," as required by applicant's independent claim 1.

Furthermore, as noted, the examiner contends that Velingker discloses:

> b) the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with one of multiple streams available on the processor. See Figs.2-3, column 5, lines 19-50, and column 6, lines 10-18. In one embodiment of the invention, the resource negotiation cell (RNC) is a register with three groups of bits, each group comprising two bits (a request bit and a completion bit). Each group is associated with a different stream. When a stream wants to request a resource or indicate access completion, a fast-write instruction including data is issued to set an appropriate bit in the corresponding group.

Applicant respectfully disagrees.

Velingker explicitly states that "[i]n particular, FIG. 4 shows in step 402 that a processor desiring to write to a shared resource first lodges a request for the shared resource by writing a predetermined logic level, e.g., a '1', to its assigned request bit in the corresponding shared resource negotiator 100" (col. 5, lines 46-50). Velingker also explains:

> **In FIG. 2, request signals from the various processors <u>are respectively latched into latches 202, 204 and 206 based on the system clock signal 108.</u> Alternatively, the shared resource negotiator may implement memory or other storage element instead of the latches 202, 204 and/or 206. Preferably, the storage element for storing the request signal from a requesting processor would be synchronous with the system clock signal 108.**
>
> **In the path of the first processor's request signal 180, arbitrating logic 297 determines that no other grant signal is active. Thus, <u>the latched request signal output from the first latch 202 is ANDed with a NOR of the grant values of all other processors, i.e., with the GRANT2 signal 192 and GRANT3 signal 194 in the disclosed embodiment.</u> The result is latched in a second latch 208, and again latched in a third latch 214, which outputs the grant value. (Emphasis added, Velingker, col. 4, lines 28-44)**

But nowhere does Velingker describe an instruction, such as a fast-write instruction, that is used to control the content of Velingker's resource negotiation register (RNR). To the contrary, controlling the content of the RNR is performed entirely through hardware implementation (e.g., using latches 202, 204 and 206) and <u>not</u> through software implementation. Because Velingker does not disclose or suggest using a fast-write instruction to control the

content of a register having multiple groups of bits each associated with a corresponding thread of multiple threads, for that reason too Velingker fails to disclose or suggest at least the features of "receiving data specified by execution of a fast-write instruction in one of multiple threads processing on the multi-threaded processor, the one of the multiple threads identified by a processing thread number, the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with a corresponding thread of the multiple threads processing on the multi-threaded processor," as required by applicant's independent claim 1.

Accordingly, applicant's independent claim 1 and the claims that depend from it are patentable over the cited art.

Independent claim 20 recites the features of "receive data specified in the fast-write instruction in one of multiple threads processing on a multi-threaded processor, the one of the multiple threads identified by a processing thread number, the fast-write instruction further specifying a register, the register having multiple groups of bits, each group of bits associated with a corresponding thread of the multiple threads processing on the multi-threaded processor." For reasons similar to those provided with respect to independent claim 1, at least these features are not disclosed by the cited art. Accordingly, independent claim 20 and the claims depending from it are patentable over the cited art.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing remarks, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good

reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

Please apply any required fees to deposit account 06-1050, referencing the attorney docket number shown above.

Respectfully submitted,

Date: Aug 1, 2007

Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21702158.doc